

# FINAL REPORT

under

AFOSR Grant No. F49620-95-1-0093

## Formal Specification and Simulation of Reference Architectures for Distributed and Safety Critical Avionics Systems

January 1st., 1995 -- March 31st., 1998

Principal Investigator: Professor David C. Luckham

Gates Computer Science Building, 4A  
Stanford University  
Stanford, CA 94305-9040  
Tel: (650) 723-1242  
Fax: (650) 723-6027  
Email: dcl@anna.stanford.edu

20000622 110

### OBJECTIVES

The original scope of this effort had two main objectives:

1. investigate fundamental implementation algorithms to extend the Rapide event-based architecture definition and simulation system by adding a capability for formal constraint-based specification of systems, and for checking conformance of systems to formal constraints,
2. demonstrate scalability of Rapide to simulate functional behavior and predict performance of different kinds of distributed systems, including avionics, simulation networks, training systems, secure information systems and command and control systems.

In addition, new directions were added to include

- a technology transition effort, to enable rapide to be used for system architecture prototyping,
- development of event -based technology, called Complex event Processing, to enable instrumentation of systems to test their conformance to design constraints.

Rapide is a new Executable Architecture Definition Language (EADL) developed at Stanford for prototyping distributed and safety critical systems containing both software and hardware components. It is the first event-based simulation language to adopt a causal event execution model. Causal event executions consist of events together with their causal relations, timing and other information. Causality and timing are partial orderings of events. Therefore, causal event executions are partially ordered sets of events (called posets). They depict distributed behavior much more accurately than the state-of-the-art simulation languages which generate sequential event traces. Rapide is also the first language to provide a rich sublanguage of event pattern descriptions that include causal relationships between events. Design and implementation of the event pattern language has been one of the focus points

of this year's research effort.

More information on Rapide can be obtained from the  
Rapide WEB Site: <http://anna.stanford.edu/rapide/rapide.html>

The work performed under AFOSR Grant No. F49620-95-1-0093  
both complemented and extended work under DARPA contracts. The AFOSR  
effort includes the following tasks:

- develop foundational algorithms for checking large Rapide poset simulations for violations of formal event pattern constraints (AFOSR).
- develop foundational algorithms for implementing Rapide hierarchical event pattern mappings to facilitate analysis of massive amount of simulation results (AFOSR and ARPA).
- develop a handbook for Rapide users of precisely defined techniques utilizing event patterns to detect common distributed systems errors, in both simulations and actual systems.  
This effort will support the transfer of Rapide technology to Airforce and DoD applications (AFOSR and ARPA).
- demonstrate scalability of the Rapide simulator and additional constraint checking tools developed under this effort, to simulate and analyze various DoD systems architectures (AFOSR).
- investigate and develop techniques for hosting Rapide EADL technology on commercial middleware so as to provide builders of distributed object systems with event-based architecting capabilities that are integrated with Java, C++ and CORBA (AFOSR and DARPA).

#### FINAL STATUS OF EFFORT RESEARCH AND TECHNOLOGY TRANSITION SUMMARY

The work carried out under our AFOSR grant has had a critical impact on our Rapide technology development and studies aimed at establishing scalability to industrial size systems and finally, transition to industry. This section will report in greater detail on several of our AFOSR-supported efforts.

Previous annual reports have described the progress and impact of this project during the various years of this research effort.  
In summary the accomplishments of this effort are:

- Rapide has been made freely available on the Rapide website an Alpha status version of the Rapide simulator, POV poset browser, Raptor Animator, and RapArch architecture construction tools. Platforms supported include Unix, Solaris and Linux. Rapide has been downloaded to approximately 300 sites worldwide. There is an active Rapide support email.
- Complex event Processing (CEP), a new technology for analyzing event generated by systems, has been developed from the the Rapide event pattern language.
- Complex event Processing has been hosted on several commercial networks and middleware products to support event-based analysis of distributed systems.

- APIs have been defined to allow the infrastructure objects supporting CEP to be interfaced with commercial middleware and networks. foundational kernel object for distributed event processing. The kernel object, called the Rapide Computation Manager, can be used as a basic component of any distributed object system that generates events in various of its component objects and distributes them to other objects of the system and/or to various system management and analysis tools. The Computation Manager Interface represents a significant step in the Rapide technology transition effort.
- We have developed a preliminary set of algorithms for implementing general event distributing objects such as the Rapide Computation Manager.
- We have developed an algorithm for implementing Rapide hierarchical event pattern mappings to facilitate analysis of massive amount of simulation results.
- We have hosted preliminary implementations of Computation Manager objects on commercial middleware such as TIBCO Rendezvous, and demonstrated their application to various Java/TIB systems.
- We have hosted preliminary implementations of Computation Manager objects on the Milstd 1553, and also on Northrop-Grumman's B2-Avionics emulation system.

#### FINAL STATUS OF EFFORT ACCOMPLISHMENTS/NEW FINDINGS

##### 1. Standards in Architecture Modelling

During 1995-96 Stanford in collaboration with TRW developed a 5-Level abstraction hierarchy for Educational and Training Systems, and applied this hierarchy to architecture modelling and performance prediction in Rapide of an Intelligent Tutor System from the University of Pittsburgh, the AdvLearn System. This work has been published and is on-going. The 5-level abstraction hierarchy was taken up by the Army (Fort Monmouth) as a possible standard for Educational and Training Systems, and is now the starting point for IEEE Standards Committee 1484.

##### 2 Event Pattern Language and Constraint Language Design

Event pattern languages will play an increasingly important role in adding new functionality to distributed event-based systems after they are fielded, and in monitoring, testing, and managing large complex software systems.

We have finished the preliminary design of the Rapide event pattern language. Rapide event patterns are specially suited to specifying the behavior of distributed and safety-critical avionics systems in terms of posets. There is a new Rapide pattern language reference manual.

##### 3 Constraint Checking Algorithms

We have developed an initial version of a set of algorithms for checking large causal event histories (posets)

generated either by Rapide simulations or by event-based distributed systems executing on commercial middleware.

#### 4 Event Pattern Mappings

We have developed an initial version of an algorithm for implementing Rapide hierarchical event pattern mappings to facilitate analysis of massive amount of simulation results.

The purpose of maps is to define how the executions of one architecture may be interpreted (or viewed) as executions of another architecture.

When a hierarchical design methodology is used to develop a low level detailed architecture from a highly abstract specification, maps relating architectures at successive levels of abstraction can be composed transitively to define maps across several levels.

Particular applications of event pattern mappings that have been established experimentally include (i) reduction of complexity of large simulations by mapping posets of events in a detailed low level simulation into single events in a higher level simulation, and (ii) runtime comparison of an architecture with a standard (or reference) architecture. Comparison of architectures is accomplished by mapping the behaviors of the domain architecture(s) into behaviors of the range architecture and checking them for consistency with the constraints of the range architecture.

The principle feature of maps is the pattern triggered rule that generates a poset. It provides the necessary expressive power needed to define correspondences between architectures. Hence the map construct described here is often called an event pattern mapping.

A map generates a new poset of events from a given poset. Essentially, a map triggers on a domain poset generated by its domain architecture(s) and generates a new range poset. The range poset may be a possible behavior of the range architecture, but is generated by the map rather than by the range architecture. Events of the range poset are causally incomparable to events of the domain poset. Dependencies between events of the range poset are induced from the dependencies between events of the domain poset.

We are now beginning to experiment with the use of maps in our various modelling and simulation experiments.

#### 5. Migration of Rapide Architecting Technology to Commercial Middleware.

Under our AFOSR effort we are investigating algorithms and techniques for hosting Rapide technology on commercial middleware such as OMG's CORBA ORBs, TRW's UNAS ORB, and TIBCO's Rendezvous. One motivation for this direction is that many new DoD distributed systems are planned to be hosted upon commercial middleware (e.g., DARPA's AITS architecture).

We have initiated a technology transition effort to develop techniques for hosting Rapide EADL technology on commercial middleware so as to provide builders of distributed object systems

with event-based architecting capabilities that are integrated with Java, C++ and CORBA. This has involved collaboration with various Silicon Valley companies including SunSoft and TIBCO.

As mentioned above, a foundational aspect of introducing Rapide causal event techniques to commercial middleware is the development of a foundational kernel object for distributed event processing. The kernel object, called the Rapide Computation Manager, will be a basic component of any distributed object system that needs to generate events in various of its component objects and distribute them to other objects of the system and/or to various system management and analysis tools. The Computation Manager is intended to provide such distributed event processing capabilities to distributed object systems hosted on middleware. Under our AFOSR sponsorship we are currently developing algorithms for a pilot implementation and feasibility study of our Computation Manager Interface.

#### 6. Application to modelling and simulation of architectures of DoD systems.

As mentioned in our FY95 and FY96 annual reports, Rapide has been applied by several organizations (Lockheed-Martin, TRW, Northrop, AeroSpace Corp), in collaboration with our DARPA and AFOSR projects, to model and simulate various DoD avionics, command and control, and Federated Simulation systems. These systems include:

- NORTHROP-GRUMMAN B2 BOMBER AVIONICS EMULATOR
- DoD ADS-HLA,
- distributed transaction management systems such as X/Open, extended beyond the present Industry standard to include security policies,
- the NSA MISSI secure architecture,
- the Navy's AEGIS weapon system (with TRW)
- DoD Computer Aided Education and Training Initiative (CAETI) systems such as the AdvLearn Intelligent Tutor,
- AirForce Satellite Ground Control Station Systems (Aerospace/TRW/USC),
- DISA's Global Information System (Lockheed/Martin)

#### APPENDIX:

Technical report on hierarchical event aggregation using CEP.

#### PUBLICATIONS

##### Technical Reports:

=====

Luckham, D.C., "Complex Event Processing in Distributed Systems", Computer Systems Laboratory Technical Report CSL--TR--98--754, (PAVG No. 78), March, 1998.

Meldal, S., Luckham, D.C., "Defining a Security Reference Architecture", Computer Systems Laboratory Technical Report CSL--TR--97--728, (PAVG No. 76), June, 1997.

Luckham, D.C., Vera, J., Belz, F., "Towards an Abstraction Hierarchy for CAETI Architectures, and Possible Applications", Computer Systems Laboratory Technical Report CSL--TR--97--727, (PAVG No. 75), April, 1997.

Madhav, N., Luckham, D.C., "Designing Reliable Programs with Rapide", Computer Systems Laboratory Technical Report CSL--TR--97--725, (PAVG No. 74), April, 1997.

Luckham, D.C., "Rapide: A Language and Toolset for Simulation of Distributed Systems by Partial Orderings of Events", Computer Systems Laboratory Technical Report CSL--TR--96--705, (PAVG No. 72), September, 1996.

#### Books/Book Chapters:

=====

Luckham, D.C., "Rapide: A Language and Toolset for Simulation of Distributed Systems by Partial Orderings of Events" in proceedings of DIMACS: Series in discrete Mathematics and Theoretical computer Science/PEDLED, DIMACS special year on logic and algorithms workshop, Princeton University, Princeton, NJ, July 24-26, 1996. Also published by American Mathematical Society.

Luckham, D.C., Vera, J., and Meldal, S., "Three Concepts of System Architecture", Submitted to CACM 1996, (Draft manuscript available, 35 pages.)

Luckham D.C., and the Rapide Design Team, "Language Reference Manuals for Rapide" (Five Manuals published 1996--1997, available by FTP at <http://anna.stanford.edu/rapide/rapide.html>).

## REPORT DOCUMENTATION PAGE

AFRL-SR-BL-TR-00-

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing this collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Project, Washington, DC 20503.

0223

Rating and reviewing  
rate for information

1. AGENCY USE ONLY (Leave blank)		2. REPORT DATE	3. REPORT TYPE AND DATES COVERED Final Technical Report 1 Jan 95 to 31 Mar 98	
4. TITLE AND SUBTITLE Formal Specification and Simulation of Reference Architectures for Distributed and Safety Critical Avionics Systems			5. FUNDING NUMBERS F49620-95-1-0093 2304/FS	
6. AUTHOR(S) Professor David C. Luckham				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Gates Computer Science Building, 4A Stanford University Stanford Ca 94305-9040			8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) AFOSR/NM 801 N. Randolph St, Rm 732 Arlington, VA 22203-1977			10. SPONSORING/MONITORING AGENCY REPORT NUMBER  F49620-95-1-0093	
11. SUPPLEMENTARY NOTES				
12a. DISTRIBUTION AVAILABILITY STATEMENT Approved for public release; distribution unlimited.			12b. DISTRIBUTION CODE	
13. ABSTRACT (Maximum 200 words) The original scope of this effort had two main objectives: 1. investigate fundamental implementation algorithms to extend the Rapide event-based architecture definition and simulation system by adding a capability for formal constraint-based specification of systems, and for checking conformance of systems to formal constraints, 2. demonstrate scalability of Rapide to simulate functional behavior and predict performance of different kinds of distributed systems, including avionics, simulation networks, training systems, secure information systems and command and control systems.  In addition, new directions were added to include  a technology transition effort, to enable rapide to be used for system architecture prototyping, development of event -based technology, called Complex event Processing, to enable instrumentation of systems to test their conformance to design constraints.				
14. SUBJECT TERMS			15. NUMBER OF PAGES 6	
			16. PRICE CODE	
17. SECURITY CLASSIFICATION OF REPORT UNCLASSIFIED	18. SECURITY CLASSIFICATION OF THIS PAGE UNCLASSIFIED	19. SECURITY CLASSIFICATION OF ABSTRACT UNCLASSIFIED	20. LIMITATION OF ABSTRACT UL	